This is **bold** and have a look at the following.

The publication subject of this page has the following reference which is the starting point to enhance our own Smalltalk class

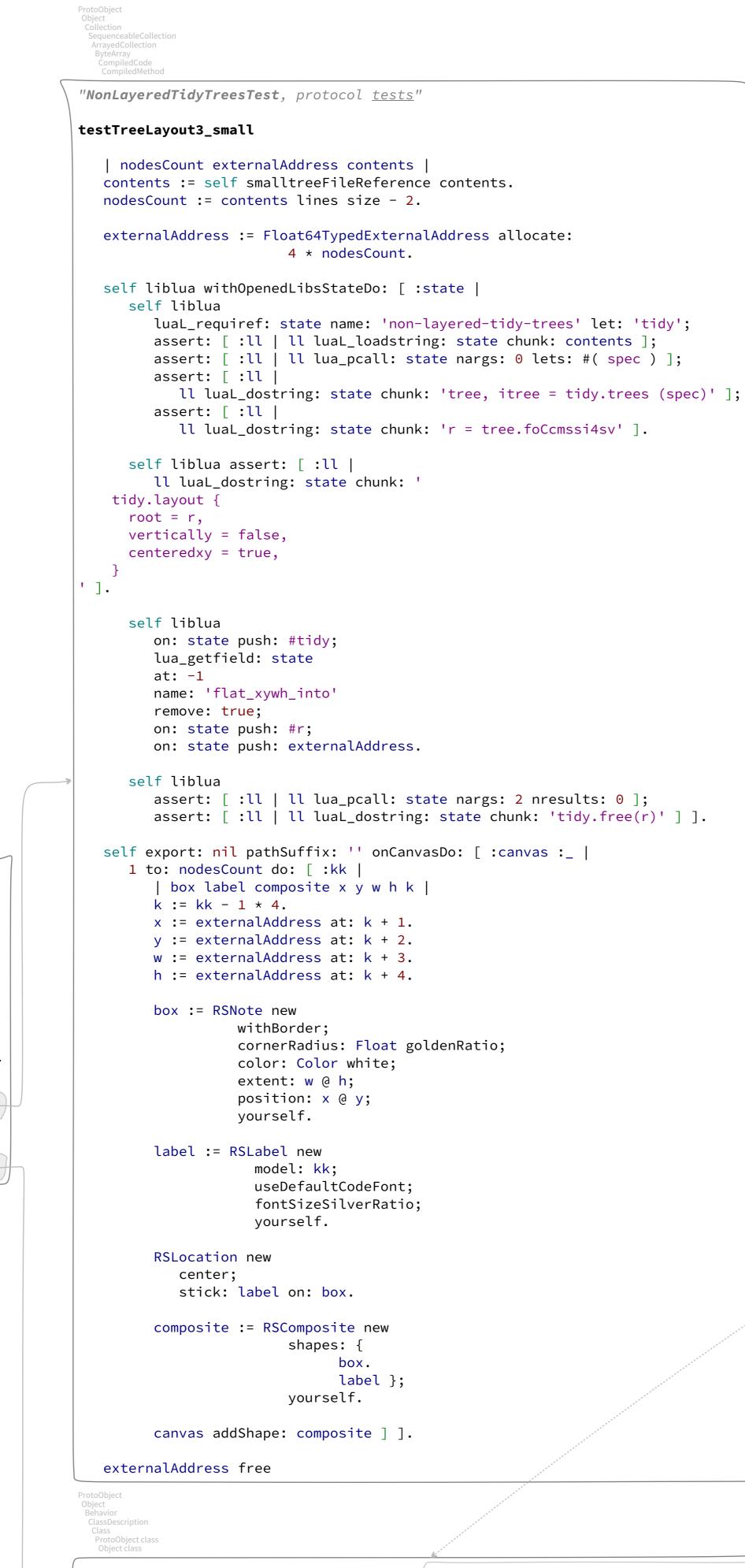
I am the vertical tree layout algorithm described in the publication "Drawing non-layered tidy trees in linear time" by Atze van der Ploeg [1] (companion repo at [2]). I use the trait RSTWalkBasedRegularTreeLayout to get the main behavior and fullfil the explicit requirements to have a tree layout growing toward the vertical dimension.

> RSVanDerPloegTreeLayout to be faster by coding the core algorithm direclty in C and using Lua as intermediate for both describing the tree nodes and interfacing with C. We represent the tree used in the paper as case study to spot the core points of the algorithms looks like either

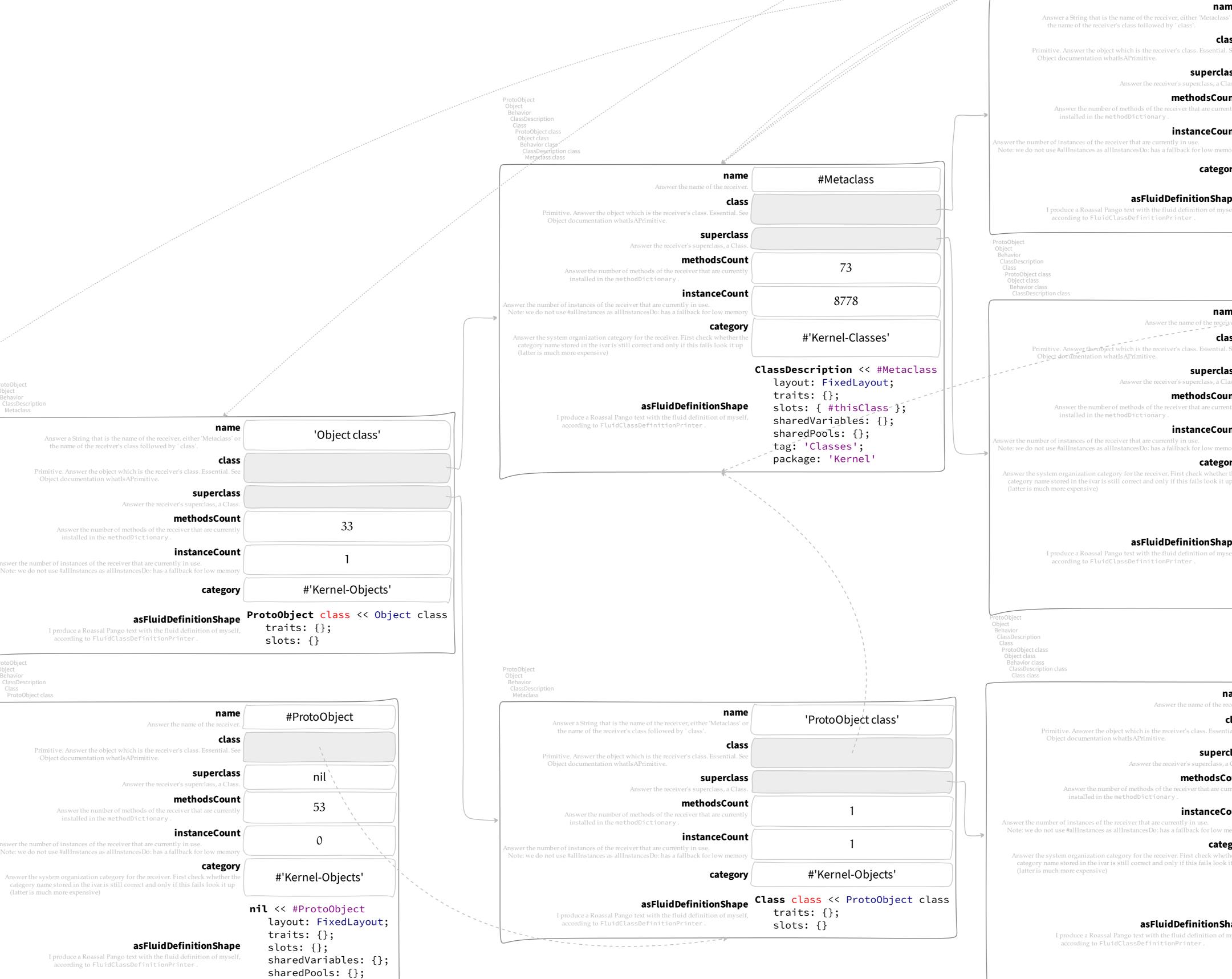
NonLayeredTidyTreesTest >> #testTreeLayout3\_small

Have a look at the Smalltalk's object model

[2]: https://github.com/cwi-swat/non-layered-tidy-trees



Answer the name of the receiver.	#Object	
class		
Primitive. Answer the object which is the receiver's class. Essential. See Object documentation whatIsAPrimitive.		
Superclass Answer the receiver's superclass, a Class.		
Answer the number of methods of the receiver that are currently installed in the methodDictionary.	471	
Answer the number of instances of the receiver that are currently in use.  Note: we do not use #allInstances as allInstancesDo: has a fallback for low memory	135	
Answer the system organization category for the receiver. First check whether the category name stored in the ivar is still correct and only if this fails look it up (latter is much more expensive)	#'Kernel-Objects'	
asFluid Definition Shape I produce a Roassal Pango text with the fluid definition of myself, according to Fluid Class Definition Printer.	<pre>ProtoObject &lt;&lt; #Object   layout: FixedLayout;   traits: {};   slots: {};   sharedVariables: { #DependentsFields };   sharedPools: {};   tag: 'Objects';   package: 'Kernel'</pre>	



tag: 'Objects'; package: 'Kernel'



#'Kernel-Classes'

asFluidDefinitionShape slots: { #subclasses . #name . #classPool . #sharedPools . #environment . #category };

Primitive. Answer the object which is the receiver's class. Essential.

installed in the methodDictionary.

superclass

methodsCount

instanceCount

#'Kernel-Classes'

Answer a String that is the name of the receiver, either 'Metaclass' or

Answer the number of methods of the receiver that are currently installed in the methodDictionary.

according to FluidClassDefinitionPrinter.

methodsCount

instanceCount

I produce a Roassal Pango text with the fluid definition of myself, traits: {};

the name of the receiver's class followed by 'class'.

Object documentation whatIsAPrimitive.

Answer the number of instances of the receiver that are currently in use.

Object documentation whatIsAPrimitive.

the name of the receiver's class followed by 'class'.

installed in the methodDictionary.

#'Kernel-Classes'

asFluidDefinitionShape ClassDescription class << Class class

slots: {}

Object documentation what Is APrimitive.

Primitive. Answer the object which is the receiver's class. Essentia

Answer the number of methods of the receiver that are currently

methodsCount

instanceCount

slots: {}

the name of the receiver's class followed by 'class'.

installed in the methodDictionary.

Primitive. Answer the object which is the receiver's class. Essential.

installed in the methodDictionary.

Answer the number of methods of the receiver that are currently

Answer the receiver's superclass, a Class.

methodsCount

instanceCount

ClassDescription << #Class</pre>

layout: FixedLayout;

sharedVariables: {};

sharedPools: {};

package: 'Kernel'

tag: 'Classes';

traits: {};

Object documentation what Is APrimitive.

methodsCount

instanceCount

#'Kernel-Classes'

Object documentation what Is APrimitive.